

A LEVEL

Exemplar Candidate Work

COMPUTER SCIENCE

H446

For first teaching in 2015

**H446/03 Summer 2017
examination series
Set C – Low**

Version 1

Contents

Introduction	3
Exemplar 9	4
Commentary	30

Introduction

These exemplar answers have been chosen from the summer 2017 examination series.

OCR is open to a wide variety of approaches and all answers are considered on their merits. These exemplars, therefore, should not be seen as the only way to answer questions but do illustrate how the mark scheme has been applied.

Please always refer to the specification (<http://www.ocr.org.uk/Images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf>) for full details of the assessment for this qualification. These exemplar answers should also be read in conjunction with the sample assessment materials and the June 2017 Examiners' Report to Centres available on the OCR website <http://www.ocr.org.uk/qualifications/>.

The question paper, mark scheme and any resource booklet(s) will be available on the OCR website from summer 2018. Until then, they are available on OCR Interchange (school exams officers will have a login for this).

It is important to note that approaches to question setting and marking will remain consistent. At the same time OCR reviews all its qualifications annually and may make small adjustments to improve the performance of its assessments. We will let you know of any substantive changes.

Exemplar 9 – Set C (Low)

Programming project (non exam assessment)

Learners will be expected to analyse, design, develop, test, evaluate and document a program written in a suitable programming language.

Project Title: Revision Program

Contents list

- Investigation description.....**pg3**
- Analysis.....**pg4**
- Design.....**pg11**
- Coded solution.....**pg15**
- Evaluation.....**pg21**
- Bibliography.....**pg26**

Problem identification / description

For my project, I am going to develop an educational quiz-based application that will help students when learning and when revising for their exams. It will be aimed at AS/A Level Psychology students, although anyone will be able to use the app if they are interested. The application will be based around quizzes that the user will complete, with questions based on the AS Level Psychology specification. The user will be given feedback on their score for each quiz, and show the correct answers and which areas the student scored lowest in. My program will then focus on these areas in future quizzes; there will be an algorithm that makes it more likely to ask questions from topics that the user struggles with, which will help the user to develop their knowledge of them. After each quiz the app will record the user's progress, so the user will be able to track their progress.

I will be using the Java programming language to develop my application. The entirety of my work will be computer-based, which has several advantages over paper-based work. Firstly, using a computer will allow for all my work to be organised easily, and I won't need to worry about misplacing sections of my project. Computers also make it much more simple to share my work, and communicate with stakeholders and clients. I can send full copies of my work to people instantly over the internet without printing or writing it out, which is a huge advantage as it saves a lot of time and resources. Programming in dedicated software is useful because I can identify where I have made code and syntax errors and adjust the code accordingly. It also allows me to test my application as many times as I like before sending it to my clients. Before beginning the development stage of my project, I will think ahead and draw up a plan of what functions I need my program to have and how to implement them. This is important to speed up the development process and make sure I always know what I want to work towards. Throughout developing the program, I will be conscious of the fact that I can re-use some parts of my code, saving myself time in the future.

Analysis

I chose to develop a mobile application because it will mean that it is very accessible for anyone that may want to use it. Almost everyone uses a smartphone every day, so having an app to help with work will be very useful. Furthermore, making a mobile app will allow me to make learning more ‘fun’; because it is in the form of a quiz, I will be able to give the user goals and targets to work towards, as well as scores for their progress. These targets will motivate the user to effectively learn, and give a sense of achievement when they do get a high score on a quiz. This is better than something like a textbook because it is more interactive, which may benefit the types of students that have difficulty engaging in textbooks. Furthermore, the app will be both free to produce and free for the users to download.

Stakeholders

Throughout the production of my project I will require several stakeholders, who will be involved in refining my program and guiding the development to tailor for the needs of my clients. I am studying A Level Psychology myself, so I have a general basis of what sort of content to include and how the app should work. It is important to have multiple perspectives, so the primary stakeholder will be a fellow Psychology student from my school. They will be a client, giving a generalisation of my target audience. My application will benefit my primary stakeholder by making revision a more enjoyable experience with frequent rewards, making studying simpler when working towards AS Level examinations.

I will also need an A Level Psychology teacher to help me with identifying relevant information from the specifications to use in my application. This is important because my project is based around producing a streamlined solution to make students more motivated to learn and to assist teachers, alongside normal Psychology lessons. My Computer Science teacher, Mr Atkinson, will have a senior management role, assisting me in the direction that I take my project and supporting the development of my program. Finally, I will be the project manager.

Limitations

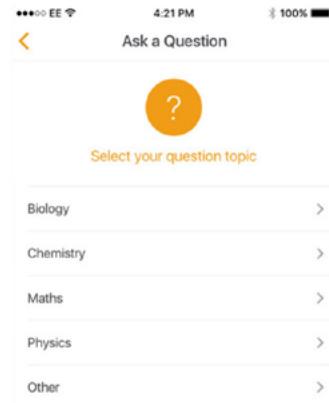
Throughout my project, there will be several limitations. Perhaps the biggest limitation will be time; I will have a limited space of time to gather data, interview my clients, and develop the solution. I was originally planning on using Java to create my program, but instead I decided to use Python instead as I am already comfortable using it. This will save me a great deal of time and will mean that I can spend as much time as possible designing and producing my solution to the specifications that I want. Nonetheless, time remains a significant limitation and I will need to work to a strict deadline. Effective time management will be necessary because I need to complete each section fully in order to complete my project.

A potential limitation is the hardware that is available to me. I am fortunate in that the devices I use at school and at home are suitable for the task at hand, and I am not expecting to run into any hardware problems throughout the duration of my project. The only requirements of my hardware are that they need to be able to run my chosen development software, and internet connectivity is required also required for several reasons. First, I need an internet connection to conduct research around my project, including learning to program in Java. I will also need to download the Eclipse software on both my school and home computers, and store backup copies of my work in the cloud. Hardware will not be an issue to me as I have an internet connection available both at home and at school, and my home and school computers can both run the necessary software.

Another possible limitation is software; some programming software can be expensive to purchase, but fortunately the Eclipse Java Development Kit (JDK) that I will be using is free, as well as Java itself. Finally, I have limited experience and expertise in programming, and I have not produced any projects on this scale before. I have programmed in other languages before, so I will be able to adapt to a new language. Nonetheless, it will no doubt take me longer than usual to code using a language of which I have limited experience, so this could be a limitation to the overall quality of my solution.

Existing solutions

A limited number of applications already exist that function similarly to the one I am developing. One such application that immediately springs to mind is Gojimo, which is also a dedicated AS Level revision app. However, an immediate drawback for this solution is that a payment is required for large sections of AS Level course content, which is a real problem for students who have a limited budget and may be less likely to use the application after discovering you need to pay to get the full benefits. This is something that I can counter with my application; all aspects of the application will be free for users to download and use.



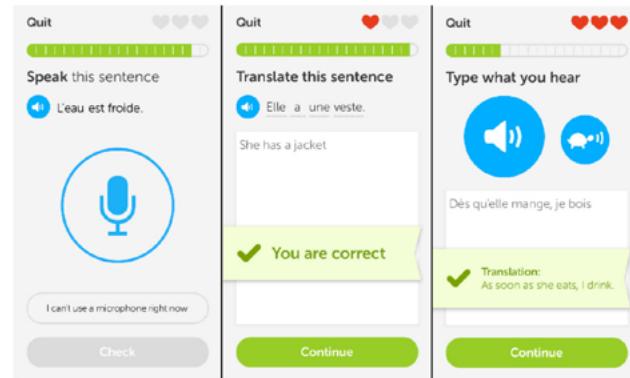
Another existing solution is Duolingo, which is dedicated for learning languages only. Unlike Gojimo, this app is completely free for users to use, and this accessibility is something that I would like to replicate.

Furthermore, I found its user interface to be far more user-friendly than Gojimo, as you can see from the screenshot to the

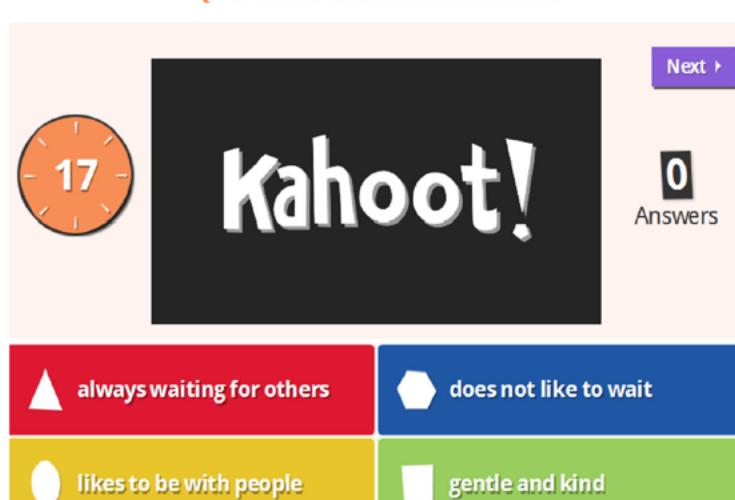
right. It made it much more straightforward to simply access quizzes and test yourself. Again, this is something that I will be attempting to mirror as I believe that most users will be wanting a simple app that makes it easy to start learning and revising. For this reason, I will only have a simple menu upon the opening of the program, with options to either select a topic or view an 'about' section to find out how to use the application.

Another advantage of Duolingo is that it can be used on multiple platforms; it is available on mobile devices and on desktop computers. This is useful because it means that the application can be used in more situations, both at home on the computer and on the go. It also maximises the audience for the app. Teenagers and adults are likely to use applications on laptops and desktop PC's, while younger children may be more inclined to use phones and tablets.

On the other hand, Duolingo has been criticized about the quizzes being inaccurate, which is something that I can make sure to avoid. Having an A Level Psychology teacher as a stakeholder will help me in this regard, because he will be able to tell me exactly what content needs to be implemented. Like I said before, Duolingo focuses purely on learning languages, so there is currently not a solution for the specific problem that my project will focus on.



Another tool that is rapidly growing in use within schools is Kahoot, which follows a very similar format to my own solution. As you can see from the screenshots below, it incorporates bright colours and shapes to be visually appealing for young children, whilst simultaneously holding a sleek, minimalist design which anyone can use from any age group and background. Another feature of Kahoot that I am fond of is the fact that you can see the collective scores of each user after each quiz. However, a rather significant limitation of Kahoot is that it does not store the user's answers after they have completed quizzes, so users cannot track their revision progress. I aim to improve this by storing user answers and giving users an option to view their previous scores.



Essential features

My solution will be based around the quiz component. Psychology-based questions will appear on the screen, and users will be able to select a multiple-choice answer. I will make sure there is a time limit for each question, meaning that questions are quick-fire. There will be a maximum of 20 questions in each quiz, varying depending on the topic. This solution is designed to be a quick way for students to test and evaluate their knowledge, and keeping the program fast-paced yet smooth will help to achieve this aim.

After each quiz, the program will display the score that the student achieved, and identify which areas need improvement. I believe this self-assessment section is crucial because it will let students see which areas of Psychology A-Level they are weakest in, and adjust their revision accordingly.

Another essential feature is a simple menu interface, which will display options to let the user select a quiz about their choice of Psychology topics. Another interesting feature that I will try to implement will be a record of the user's results history. This would be a very useful feature because it would mean the user can track their progress, and gauge a sense of how their revision has helped enhance their overall knowledge. I have considered branching this out and including some form of online leader board, adding a competitive element to revision. However, this will be problematic to develop and due to time constraints, and the fact that I have limited programming experience, I must leave this out.

Success criteria

My solution will in the form of an interactive quiz-based program. The core function will be to present a sequence of questions in a quiz, allowing for users to select answers one by one in a multiple-choice basis. I have split my success criteria into three sections;

1. **My solution must incorporate all my essential features.** It is important to me that my final product is loyal to my initial idea of the program. Inevitably, I will find during the development process that some of my features are difficult or even impossible to effectively implement. If I can form reasonable compromises and make sure my program functions in the way I want it to, I don't see this as a problem. In fact, there is a good chance that I will realise the user will benefit from a completely new feature that I didn't think to include before starting development. I believe that being spontaneous in this project is invaluable, and being too rigid could negatively impact the overall ingenuity and accessibility of my solution.
2. **My solution must be beneficial for the target audience and enjoyable to use.** Being user-friendly and accessible is of the upmost importance to me, as the program is aimed at a wide audience. I want to make sure that everyone from different backgrounds and age groups can use it and have a genuinely enjoyable learning experience, whether people are using the application to revise for examinations or just feel like learning something new.
3. **My project must be completed to a high standard, within the given time frame.** As I mentioned before, time is a big limitation. I need to ensure that time is not a hindrance to the quality of my solution. This criteria point will be fulfilled when I reach a point at which I am satisfied with the functionality and quality of my program.

Investigation

To develop an effective application, it is necessary to investigate what is required and what features would be useful. I will do this by conducting an interview with my A Level Psychology teacher, which will help me gather the content for use in the application. I will also create a questionnaire to hand out to Psychology students, finding out what features the students are most interested in for a revision app, and what would motivate them to use an interactive revision tool. I will make sure to be polite and respectful while carrying data, but the interview and questionnaires will be relatively relaxed and informal.

Below is a copy of the questionnaire that I will give to Psychology students. While the answers given by students will not contain any sensitive or personal information, I will ask students not to write their name on the questionnaire, and assured them that their answers will remain anonymous and confidential, in order to avoid complications with data collection.

Psychology revision tool questionnaire for students

Your answers will be used to help me develop a computer-based revision tool. Please do not include your name or any other personal information, to maintain anonymity and confidentiality. You may withdraw your data at any time, during or after the questionnaire. Thank you.

1. Do you currently use any digital revision tools? If so, which?
2. Do you think you would benefit from using a digital revision tool?
3. What features would you expect from a computer-based revision program?
4. How much time do you spend revising in a single revision session?
5. Would you use an online revision tool alongside normal revision, or instead of?
6. Which topics in the Psychology course do you struggle with the most? (for example, Memory or Forensic Psychology)

If there is anything else you would like to mention or suggest, please write it below. Thank you for participating.

I will also conduct an interview with my Psychology teacher, but since I will only be gathering information about aspects of Psychology to do with units and topics in the course, I don't feel that it is necessary to form a transcript, as the interview is not directly related to the design of my application, but rather the content of the quizzes.

The Psychology questions themselves are a pivotal aspect of my program, I will need to allocate a significant amount of time for writing up my questions in a separate word file before implementing them into my coded solution. Also, I plan to have questions from each quiz in text files outside of my main code, with a file for each set of questions. My program will then import each set of questions, and the file that is imported depends on which topic the user chooses. This will make my code far more neat and tidy than it would be if I had to write out each question in the code itself, as I will need to have 70 questions, on top of other text messages.

Design

Stepwise refinement

Below are the stages that the user will navigate through when using my program. They include the core functions of the program, and the choices that users will make. I have broken down each stage into smaller sections, iterating what will happen and accounting for the possible outcomes that the user will choose.

0. Quiz
1. Menu screen – select a topic
2. Ask questions to user
3. Collect data
4. Evaluate data
5. Give user feedback
6. Start quiz again/end program

1.

- 1.1** Display a list of Psychology AS Level topics for the user to select a quiz on.
- 1.2** Print a message asking the user to select a topic (each topic assigned a number)
- 1.3** Each topic name corresponds to a set of quiz questions
- 1.4** Start the quiz that the user chooses

2.

- 2.1** Bank of Psychology questions
- 2.2** Display questions to user sequentially, at random
- 2.3** Display a set of four multiple choice answers, with one correct answer.
- 2.4** Allow a certain amount of time for the user to select an answer (15 seconds)

3.

- 3.1** Ask user for the input of answers after each question
- 3.2** Store each question answer as a variable (multiple choice answer)
- 3.3** If no input is received after 15 seconds, move on to the next question

4.

4.1 Compare each answer to the correct answer from the bank

4.2 Create a score for the user from their given answers

5.

5.1 Display the user's score

5.2 Print a comment about the user's results

5.3 Show the user the answers to any questions they got wrong

6.

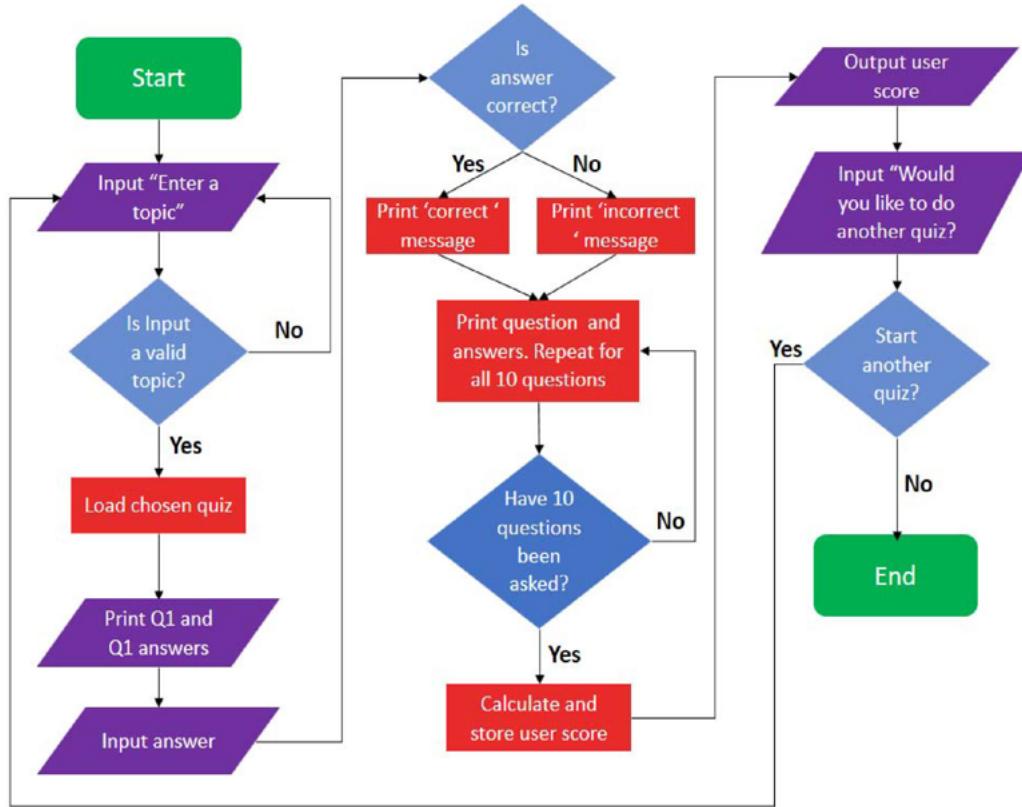
6.1 Ask the user if they would like to complete another quiz.

6.2 If they say yes, loop the program from the beginning.

6.3 If they say no, end the program.

Flowchart

Here is a flowchart showcasing everything the user will experience when navigating my program, from start to finish.



My flowchart is based on pseudocode. By following the flowchart, you can gain an idea of the user experience as they navigate my program. Being a flowchart, it is a drastically simplified version of the program. The process sections of the flowchart will be fleshed out into fully working functions, and the decision tiles will be replaced with sets of 'if' and 'elif' statements in the final code.

My program will be heavily based around decisions from the user, which you can see from the number of decision and input tiles in the flowchart. The user experience is likely to be different from user to user, as different inputs change what the user sees. For example, near the beginning of the program, shortly after 'start', the user decides which quiz topic they would like to do. The quiz itself is also based around decisions and inputs from the program user. The main function is the quiz section, making up the bulk of my program. In the flowchart, I also showed the way in which the loop will work. The user will be given the opportunity to loop at the end of each quiz, and doing so will loop the program back to the menu screen, so that the user can start another quiz.

Algorithms

Due to the way that my program will follow a linear structure for the user, and will not necessarily need to process complex calculations. The primary algorithm is the quiz; giving the users questions and answers from external text files, and comparing inputs to the correct answers. Another algorithm will be used after the user has completed a full set of questions in a quiz. My solution will add the scores together and calculate the total score for the user's chosen Psychology topic. The program will then compare the users score to a list of possible score outcomes, and depending on how many points they earned, different result messages will be output for the user to read.

Another algorithm will be used to determine whether all 10 questions have been given. If 10 questions have not been asked, the application will continue to ask the user questions. If all questions have been asked, the program will stop asking questions and move on to calculate and output the user's score. I also aim to implement a randomising variable, so the user is given questions in a random order instead of the same order every time. This is good because it forces users to learn the actual answers to each question, as opposed to learning the answers in the order in which they are presented. The randomisation will occur when the code is importing questions from external text files; I will write the code in such a way that the program imports a random line each time. Of course, I will therefore also need to make sure that the same question is not imported twice, because it is obviously counter intuitive to ask users the same question more than once in a quiz. To solve this, I will write an algorithm that compares the randomly selected question to the ones that have already been asked. If the question has not been asked, it will print the question to the user and ask for input. If it has been asked, the program will run the randomisation algorithm again and select another question, repeating until all questions have been asked.

Once all questions have been answered, the program will move on to the 'debrief' section, where the user can either choose another quiz (looping the program from the beginning using a while loop) or give the option to end the program.

Key variables

I will be using a while loop in my program, so the user can choose to start another quiz if they choose. For this loop, I will use the variable 'Finish', which I will declare at the start of my code. This variable will be set to the value '0' throughout the duration of my program running, unless the user chooses not to start another quiz, in which case the Finish variable will be set to '1', and the program will end.

Each answer that the user enters for each question will be stored as a unique variable, for use when comparing to the correct answers. The program will then calculate a total score, which will also be stored as a variable and will be output to the user.

Coded solution

While writing the code for my program, I tried to follow my flowchart closely and ensure that it all flowed smoothly. The flowchart version is obviously much simpler than the final product, but it still ended up working the way that I hoped. However, I did have to make some sacrifices along the way. Due to my limited time for the coding stage, I was forced to drop some of my initial features that I wanted and form a compromise with the time that I had. Nevertheless, my program still works how I want it to, and I managed to eliminate bugs and other problems, which I will elaborate on in the testing section.

I chose to work in chronological order, laying out the code in an order that is simple to follow. The first thing that I did was create my menu screen that the user would see upon opening the program.

```
import time #This allows time to be used in the program.
import random #This allows randomisation to be used.
import sys #This allows the user to safely close the program.

loop = 0
score = 0
questionCounter = 0
x=0

while x==0: #This is the start of the while loop, which will give the user the option to loop the program if they want to.

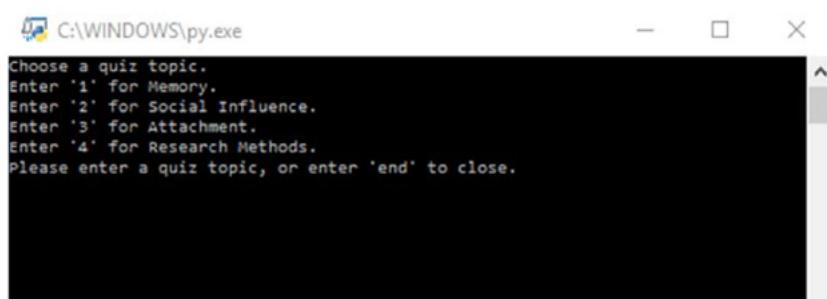
    print ("Welcome to my Psychology A Level revision tool.")
    time.sleep(1)
    print("Choose a quiz topic.")
    time.sleep(1)
    print("Enter '1' for Memory.")
    time.sleep(0.5)
    print("Enter '2' for Social Influence.") #This section is a simple menu screen, asking the user to input a study topic.
    time.sleep(0.5)
    print("Enter '3' for Attachment.")
    time.sleep(0.5)
    print("Enter '4' for Research Methods.")
    time.sleep(0.5)

    choice = input("Please enter a quiz topic, or enter 'end' to close.")


```

The 'choice' variable stores the input from the user. The value of the variable determines which quiz is loaded. You can also see that I imported time, random and sys into my code; I have explained the function of these in the red comments in my code. I used time.sleep in the menu screen to purely make the program to seem more fluid and professional when the user is navigating it. In this section I also started the while loop, which as I mentioned in the comments, gives the user the option to start another quiz after finishing one, or close it. I also set several variables to the value 0, which are used later in the code.

This is a screenshot of the menu screen as it appears to the user.



C:\WINDOWS\py.exe

```
Choose a quiz topic.  
Enter '1' for Memory.  
Enter '2' for Social Influence.  
Enter '3' for Attachment.  
Enter '4' for Research Methods.  
Please enter a quiz topic, or enter 'end' to close.
```

Before starting to write the next portion of code, I had to create a text file for each set of questions, and each set of possible answers, including a separate file for each set of correct answers for every question. Before making these files, I wrote out my questions and answers in a word file, which was much easier to work with. Creating the text files was just a process of copying and pasting the questions and answers.

An alternative to this would have been to write out every question and answer within the code itself, but this would make my code extremely cluttered. For the sake of tidiness, I chose to store everything in text files, and import information from these files rather than writing it all out in my code. I have attached a picture of the folder containing the text files below, as well as a screenshot of the original word file. Having so many text files was necessary, since I made a file for each set of questions and answers in each quiz (of which there are 4).

Computing coursework – Psychology AS Level questions and answers

QUIZ 1 – Memory

1. What are the two memory models called?
 - a. Multi-save memory model and working memory model
 - b. Multi-store memory model and multi-operating model
 - c. **Multi-store memory model and working memory model**
 - d. Multi-save memory model and active memory model

2. Which type of long-term memory is responsible for storing information about events?
 - a. **Episodic memory**
 - b. Procedural memory
 - c. Semantic memory
 - d. Agentic memory

3. In the working memory model, which memory system handles visual and spatial information?
 - a. Episodic buffer
 - b. Phonological loop
 - c. **Visuo-spatial sketchpad**
 - d. Central executive

Name	Date modified	Type	Size
quiz1Aanswers	17/03/2017 15:00	Text Document	1 KB
quiz1Banswers	17/03/2017 15:07	Text Document	1 KB
quiz1Canswers	17/03/2017 15:09	Text Document	1 KB
quiz1CorrectAnswers	17/03/2017 14:46	Text Document	1 KB
quiz1Danswers	17/03/2017 15:10	Text Document	1 KB
quiz1Qs	13/03/2017 11:57	Text Document	1 KB
quiz2Aanswers	20/03/2017 09:50	Text Document	1 KB
quiz2Banswers	20/03/2017 09:52	Text Document	1 KB
quiz2Canswers	20/03/2017 09:53	Text Document	1 KB
quiz2CorrectAnswers	17/03/2017 14:48	Text Document	1 KB
quiz2Danswers	20/03/2017 09:55	Text Document	1 KB
quiz2Qs	13/03/2017 12:06	Text Document	1 KB
quiz3Aanswers	20/03/2017 09:56	Text Document	1 KB
quiz3Banswers	20/03/2017 09:58	Text Document	1 KB
quiz3Canswers	20/03/2017 10:00	Text Document	1 KB
quiz3CorrectAnswers	17/03/2017 14:50	Text Document	1 KB
quiz3Danswers	20/03/2017 10:02	Text Document	1 KB

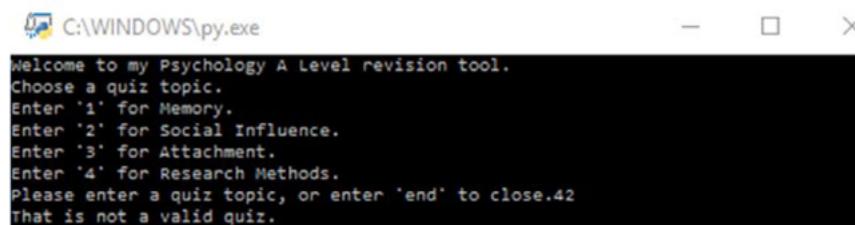
The next section of my code simply reacts to the user's input about their quiz choice, and loads the corresponding quiz. In between this section is a smaller section that simply prints the user's choice.

```
if choice == ("1"):
    #This section opens the text files for each question and answer of a quiz.
    questionList = open("quiz1Qs.txt", "r").readlines() #The set of text files opened depends on the user's choice.
    correctAnswer = open("quiz1CorrectAnswers.txt", "r").readlines() #I created text files for each answer and question list in each quiz.
    answerA = open("quiz1Aanswers.txt", "r").readlines() #I gave the files simple names to make it more straightforward.
    answerB = open("quiz1Banswers.txt", "r").readlines()
    answerC = open("quiz1Canswers.txt", "r").readlines()
    answerD = open("quiz1Danswers.txt", "r").readlines()

elif choice == ("2"):
    questionList = open("quiz2Qs.txt", "r").readlines()
    correctAnswer = open("quiz2CorrectAnswers.txt", "r").readlines()
    answerA = open("quiz2Aanswers.txt", "r").readlines()
    answerB = open("quiz2Banswers.txt", "r").readlines()
    answerC = open("quiz2Canswers.txt", "r").readlines()
    answerD = open("quiz2Danswers.txt", "r").readlines()
```

If the user chooses the second quiz, all the questions and answers will be loaded from that quiz. At this point, it was very important that I had simple names for the text files and the variables, as overcomplication could easily lead to becoming confused and making it harder to load the correct questions and answers.

This section only opens the question lists and answers for each quiz, but doesn't do anything with them at this point. The next section of code shows how the questions and answers are used and how the quiz is formed and presented to the user. As well as using if statements and elif statements for the user's 'choice' input, I also made sure to include an elif statement that reacts to the user entering something irrelevant, such as '55' or 'banana'. This ensures my code is robust and durable, and gets rid of any bugs that may be caused by bad input otherwise. Below is a screenshot of my application reacting to invalid input from the user. After telling the user the input is wrong, the menu screen loops from the beginning so that the user can enter another input.



Welcome to my Psychology A Level revision tool.
Choose a quiz topic.
Enter '1' for Memory.
Enter '2' for Social Influence.
Enter '3' for Attachment.
Enter '4' for Research Methods.
Please enter a quiz topic, or enter 'end' to close.42
That is not a valid quiz.

Below is the part of my code that outputs questions to the user, takes input and calculates the user's score.

```
usedIDs.append(questionID)
currentQuestion = questionList[questionID]
currentA = answerA[questionID]
currentB = answerB[questionID]
currentC = answerC[questionID]
currentD = answerD[questionID]
print("")
print("Question: "+currentQuestion)
time.sleep(1)
print("A. "+currentA)
print("B. "+currentB)
print("C. "+currentC)
print("D. "+currentD)
time.sleep(1)
answerGiven = input(str("Please enter your answer."))
if answerGiven in correctAnswer[questionID]:
    score+=1
    print("Correct.")
else:
    print("Incorrect.")
printScore=str(score)
print("Your current mark is "+printScore+".")
questionCounter+=1
if questionCounter == 10:
    time.sleep(1)
    print("You have completed the quiz. Calculating your score.")
    time.sleep(1)
    print("...")
    time.sleep(1)
    print("Your score is:",score)
loop = 2
time.sleep(1.5)
```

The 'answerGiven' variable takes the user's answer and compares it to the correct answer, which is stored in the 'correctAnswer' text files. I used an if statement; if 'answerGiven' is equal to the 'correctAnswer' for any given question, a value of 1 is added to 'score'. This value gets higher as the user answers questions correctly, with the maximum score being 10, if the user manages to answer every question correctly in a quiz. The current score is also printed to the user after every question, and the user is immediately told whether they answered the question right or wrong.

When testing my program, I found a bug involving the score continuing to go up even when the user started a new quiz. For example, if the user took a Social Influence quiz and got a score of 7, and then took an attachment quiz and scored 8, my program would display the score as 15. This was not ideal, because it meant that the user could not track their score progress if they chose to loop the program and complete another quiz. To get around this, I wrote a piece of code that resets the user's score if they chose to complete another quiz.

```
restart = input("Would you like to start another quiz? [Y/N]")
if restart == "Y": #If the user chooses yes, the program will loop and they can complete another quiz.
    x=0
    mark=0 #This resets the mark and the score so that the program has a new score for each quiz.
```

This made my code more robust. It was something that I didn't notice until I got one of my stakeholders to test my program, so having a stakeholder was invaluable to me in the development and testing of my code.

I have included a screenshot of my quiz running below. You can see that it works how I intended. The program takes the user's input and identifies whether the answer was correct or incorrect. My 'score' feature also works exactly how I wanted; a correct answer adds 1 mark, and an incorrect answer doesn't alter the mark.

```
question: Which one of these was a limitation of the leading questions car crash study?
A. Small sample size
B. Individual differences
C. Gender bias
D. Reductionism

Please enter your answer.A
Incorrect.
Your current mark is 0.

Question: Which famous psychologist proved that leading questions can have a distorting effect on memory and recall?
A. Loftus
B. Bowlby
C. Theroux
D. Jones

Please enter your answer.A
Correct.
Your current mark is 1.
```

```
Question: Which one of these was a limitation of the leading questions car crash study?  
A. Small sample size  
B. Individual differences  
C. Gender bias  
D. Reductionism  
  
Please enter your answer.A  
Incorrect.  
Your current mark is 7.  
You have completed the quiz. Calculating your score.  
...  
Your score is: 7  
Would you like to start another quiz? [Y/N]
```

Above is a screenshot of what the user sees once they have completed all questions in the quiz. The user is told that they have finished the quiz, and their total marks are added up and printed to the user. In the screenshot, you can see that I answered 7 out of the 10 questions correctly, so therefore my score is 7.

The program also creates a 'results' file in the same file in which the questions and answers are scored, as well as the user's score value. This is a very important feature that I wanted to include, because it means that users can see their correct and incorrect answers, helping them to get a better understanding of parts of the Psychology A Level course that they may not be comfortable with yet. The program writes to the newly created results file and stores the user's score information, and it does this as many times as the user wants. For example, they may want to view the results of two consecutive quizzes that they complete, and the program will create and write to two separate results files.

After the quiz has been completed, the user has the option to loop the program and do another set of questions, or to close the quiz. Below I have included a screenshot of the program starting from the beginning, as I feel this is a very important feature to have.

```
Please enter your answer.C  
Correct.  
Your current mark is 8.  
You have completed the quiz. Calculating your score.  
...  
Your score is: 8  
Would you like to start another quiz? [Y/N]Y  
Welcome to my Psychology A Level revision tool.  
Choose a quiz topic.  
Enter '1' for Memory.  
Enter '2' for Social Influence.  
Enter '3' for Attachment.  
Enter '4' for Research Methods.  
Please enter a quiz topic, or enter 'end' to close.
```

On the other hand, if the user chooses not to start another quiz, the program thanks the user for using the application and safely closes. This is the code for this end section of my program. 'x' is set to 1, which stops the program looping and allows it to move on to

sys.exit, which safely closes the program. Before this, the results file is closed, and the user can navigate to this file and see their answers for each question.

```
results.close()

restart = input("Would you like to start another quiz? [Y/N]")
if restart == "Y": #If the user chooses yes, the program will loop and they can complete another quiz.
    x=0
    mark=0 #This resets the mark and the score so that the program has a new score for each quiz.
else:
    x=1
    print("Thank you for using my quiz. Closing...")
    time.sleep(1.5)
    sys.exit() #This closes the program safely.
```

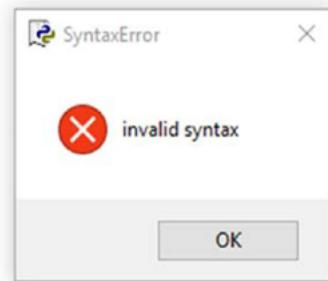
Testing

Throughout the whole development process, I was constantly testing newly written parts of my code as I was coding. There were some major bugs that had to be ironed out after I had finished most my program, but I found it more productive to make sure my code worked before moving on. I complied a set of screenshots for the testing process, showing the problems I faced and how I identified and fixed the errors within my code.

One of the first errors I encountered was in the menu screen. I ran into a syntax error when taking the user's input about their quiz choice. When I tried to run my program to check that it was taking and manipulating the input in the way that I wanted it to, I was notified by IDLE that my syntax was invalid.

```
choice = input("Please enter a quiz topic, or enter 'end' to close.")

if choice == ("1"):
    print ("Loading the Memory quiz.")
    time.sleep(1)
    loop = 1
elif choice == ("2"):
    print ("Loading the Social Influence quiz.")
    time.sleep(1)
    loop = 1
elif choice == ("3"):
    print ("Loading the Attachment quiz.")
```



Fortunately, it was a very minor error, and rectifying it was just a case of making I had two equals signs in each if and elif statement. After fixing my syntax, my menu screen worked perfectly. This error showcases a strong advantage of IDLE (the integrated Python development environment); it doesn't run code that contains syntax errors, and highlights the type of error and where the error was made. This was helpful to me because it saved me time when identifying problems in my code, and gave me more time to spend on developing more code.

Another problem that I ran into was the player's score continuing to count if the player did more than one quiz at a time. This was not an error with my code, but rather a case of me forgetting to reset the mark after a quiz was completed and the score was stored in a results file. I mentioned this issue in the development section of my write up, and I only realised this bug happened when I asked one of my stakeholders to try out my program. Below is a screenshot of what happened to the score when the user completed more than one quiz, prior to me adding the score reset.

```
Question: Who conducted the Stanford prison experiment?  
A. Stanford  
B. Milgram  
C. Asch  
D. Zimbardo  
  
Please enter your answer.D  
Correct.  
Your current mark is 13.  
  
Question: What is the name of a concept concerned with internal control versus external control?  
A. Locust of control  
B. Locus of control  
C. Locus of power  
D. Grasshopper of control  
  
Please enter your answer.B  
Correct.  
Your current mark is 14.
```

This was obviously a significant problem because it meant that one of my key features did not work every; it only worked the first time the user completed a quiz. I needed to improve the feature to make the program more durable. As previously mentioned, adding a line of code to set 'mark' to 0 made sure that the score was back to 0 at the start of a new quiz. The score is only reset if the user enters 'Y' and opts to do another quiz. If they enter 'N', the program closes.

```
results.close()  
  
restart = input("Would you like to start another quiz? [Y/N]")  
if restart == "Y": #If the user chooses yes, the program will loop and they can complete another quiz.  
    x=0  
    mark=0 #This resets the mark and the score so that the program has a new score for each quiz.  
else:
```

Another area that causes problems when developing my solution was in the section of my code that opened each text file for the questions and answers in each quiz. While I tried to keep it as straightforward as possible with simple text file and variable names, I did experience problems with opening the text files. For example, I initially had a problem where only the questions and answers were loaded from the first quiz, even if the user chose a completely different one. Problems like these were primarily caused by my lack of experience of importing files in Python; I essentially had to teach myself aspects of coding in Python, and it was often a case of trial and error in getting bits of my code work.

I also found that the results file did not get created unless it was in the same folder as the text files, but unfortunately due to time constraints I did not have time to find a way around this. It remains a minor problem, nonetheless.

Evaluation

I am pleased with how my final coded solution turned out. This project has been the largest and most detailed program I have created, and I have created a program that not only works well, but will also be genuinely useful for many people. Having a revision program tailored specifically to my Psychology A Level specification is something that will really benefit myself and other Sixth Form students when studying and revising for examinations. Furthermore, it will continue to be useful in the future, for other Psychology students that are currently in lower years in school. I will pass on my program to the Social Sciences department at my school, who can then advise students about using it to aid revision. My code will remain open-source; since my code is easy to understand for anyone who is experienced with Python and programming, anyone can modify my code to either add other quizzes, or add different features and personalise it as they please.

My success criteria for this project was partially met. I managed to stick to the flowchart very well, and the user experience is exactly how I planned. The core functions of my program work flawlessly. My menu screen is concise and easy for the user to navigate. It is robust, in that any invalid input simply causes the program to tell the user their input was not valid, and loops the program from the beginning. Users can also choose to close the application if they wish. The quiz asking and answering section also works as I planned, though this section took me by far the longest to implement as I had to teach myself how to import and manipulate text files in Python.

However, there are a few features that I was forced to abandon due to the lack of time. In my Analysis section of this project, I considered forming some sort of leader board, possibly online, so that users can compare their scores and provide further motivation for studying. However, after researching this feature I came to the realisation that it would be extremely difficult to this, especially in Python. As a compromise, I implemented a more undemanding results feature, which stores the user's results as I mentioned before. Admittedly, this feature is more simplistic than I would like. The user's score is stored in a newly created text file, but you cannot see the answers that were given by the user, or the correct answer. If I was given more time to work on this project, I would work to improve this feature, with the addition of better user feedback.

The biggest limitation in this project was the time constraints. With more time, I would be able to build upon my existing solution, expanding and enhancing the user experience. Another limitation, linked to limited time, is the fact that I needed to teach myself more advanced Python programming techniques before starting to code my solution. If I already knew exactly what to do after planning the layout and key features, I would have saved a lot of time, which would mean I could spend much more time refining my solution, in ways that I talked about above. On the other hand, I anticipated that there could be limitations to do with my available hardware. As it turned out, there were no problems at all. The entirety of my work was completed on my laptop and on school computers, which were both

completely fine. There were also no problems in sending my work around on my home internet connection and on the school network.

I have several recommendations for anyone who is planning to undertake a project such as my own. Firstly, I would recommend gaining a thorough understanding of the programming language in which you are going to be coding your solution. This will eliminate any problems involving a weak knowledge of the language, and lets you develop your solution exactly how you want it to be. I would also recommend having someone else constantly testing your program with you and providing constructive criticism of your work. I did this by working with a stakeholder, who helped me determine which features were necessary and what content was important to include in the quizzes. I am satisfied with my final product, and it has the potential to be very beneficial to myself and other students, which was ultimately my goal when I set out to complete this project.

Bibliography

- Gojimo revision app - <http://www.gojimo.com/>
- Kahoot learning – <https://kahoot.it/>
- Codecademy - <https://www.codecademy.com/learn>
- After hours programming - <http://www.afterhoursprogramming.com/>

Examiner commentary

Question/Part: AO 2.2 Analysis

Marks: 4/10

The project concept is introduced, and as the URS acknowledges, the concept is simple. A limited attempt to Computational thinking is attempted but only in terms of why a computational solution is suitable rather than covering relevant points from section 2.2.2 of the H446 specification.

The identification of stakeholders gives detail of who will be involved in developing the solution and how they will make use of it.

The existing quiz solutions are analysed and features picked out to bring forward to the essential features of the solution. Further investigation is planned at the end of the section but no evidence that this was carried out is included.

The essential features are explained simply, most of these are measurable. Some general success criteria are discussed, although these are very broad and it is not clear how each will be measured.

Several limitations are explained, although these are fairly generic instead of aspects or features of the potential solution that have been identified as not being solvable.

The teacher's mark is appropriate.

Some comments are present in the limited code shown. Variables used appear to be sensibly named.

There is no evidence of any validation of the solution.

The work just meets the descriptors for mark band 1. The mark of 5 is a little lenient.

Question/Part: AO 3.2 Testing to inform development

Marks: 3/10

Alongside several issues described and evidenced in the write-up section, the testing section contains one test carried out and fixed during development.

This is just enough to agree the mark given.

Question/Part: AO 3.3 Testing to inform evaluation

Marks: 1/5

The second test shown in the Testing section shows a logical error was found and remedied post-development. A further test for functionality is described although there is no evidence of it being carried out.

The mark of 1 can be agreed although evidence is very limited.

Question/Part: AO 3.1 Design

Marks: 3/15

The stepwise refinement section shows a basic understanding of how the problem decomposes into smaller sections.

A system flowchart gives an overview of the solution as a whole. There is an algorithms section present, however this is a prose description of how the program will work; no flowcharts, pseudocode or structured English breakdown of algorithmic steps are included.

There is no attempt to plan any GUIs or address usability features.

Two variables are identified but this is clearly not representative of all key variables that will be part of the solution. No data structures or validation is identified (although in a multiple choice quiz no user input except a click may be needed).

No test strategy or data are present.

Question/Part: AO 3.3 Evaluation of solution

Marks: 4/15

The evaluative comments are basic and not reflective of how simple the solution is, although there is some discussion of this in the 3rd paragraph of the section. The student has not evaluated against each of the requirements identified in the Analysis section and there is no link to any evidence in the project of those partially achieved.

An attempt to discuss limitations is present but again does not cover the range of features not implemented.

The mark awarded could be seen as lenient as so few of the objectives have been commented upon, however it is only slightly generous and can be agreed.

Question/Part: AO 3.2 Developing the coded solution

Marks: 5/15

There is some evidence of a limited solution.

Code presented seems to be procedural/ linear in nature and runs in a simple text-based editor. Simple loops and selection statements are used in conjunction with text files to perform a basic quiz and keep a score per quiz. This is reminiscent of a GCSE level task and does not have the expected scope for A Level.



We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

Whether you already offer OCR qualifications, are new to OCR, or are considering switching from your current provider/awarding organisation, you can request more information by completing the Expression of Interest form which can be found here:

www.ocr.org.uk/expression-of-interest

OCR Resources: the small print

OCR's resources are provided to support the delivery of OCR qualifications, but in no way constitute an endorsed teaching method that is required by OCR. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

This resource may be freely copied and distributed, as long as the OCR logo and this small print remain intact and OCR is acknowledged as the originator of this work.

OCR acknowledges the use of the following content:
Square down and Square up: alexwhite/Shutterstock.com

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications:
resources.feedback@ocr.org.uk

Looking for a resource?

There is now a quick and easy search tool to help find **free** resources for your qualification:

www.ocr.org.uk/i-want-to/find-resources/

www.ocr.org.uk/alevelreform

OCR Customer Contact Centre

General qualifications

Telephone 01223 553998

Facsimile 01223 552627

Email general.qualifications@ocr.org.uk

OCR is part of Cambridge Assessment, a department of the University of Cambridge. *For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored.*

© OCR 2017 Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee. Registered in England. Registered office 1 Hills Road, Cambridge CB1 2EU. Registered company number 3484466.
OCR is an exempt charity.



Cambridge
Assessment

